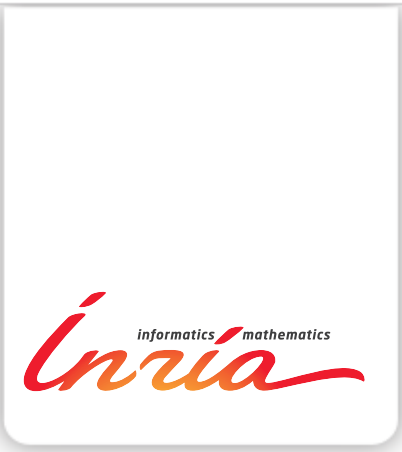

# AntDroid: A distributed platform for mobile sensing

**Nicolas Haderer[1,2], Romain Rouvoy[1,2], Lionel Seinturier[1,2,3]**
**[1] INRIA Lille – Nord Europe, Project-team ADAM**
**[2] University Lille 1 - LIFL CNRS UMR 8022, France**
**[3] Institut Universitaire de France**

**RESEARCH REPORT**
**N° 7885**
February 2012
Project-Teams ADAM

ISSN 0249-6399 ISRN INRIA/RR--7885--FR+ENG




# AntDroid: A distributed platform for mobile sensing

Nicolas Haderer[1,2], Romain Rouvoy[1,2], Lionel Seinturier[1,2,3]
[1] INRIA Lille – Nord Europe, Project-team ADAM
[2] University Lille 1 - LIFL CNRS UMR 8022, France
[3] Institut Universitaire de France

Project-Teams ADAM

Research Report n° 7885 — February 2012 — 27 pages

**Abstract:**
Scientific communities extensively exploit simulations to validate their theories. However, the relevance of the obtained results highly depends on the accuracy of the dataset they use. This statement is particularly true when considering human mobility traces, which tend to be highly unpredictable. In this paper, we therefore introduce ANTDROID, a distributed platform for sensing the activities of mobile users. In particular, ANTDROID offers to scientists a participative platform for helping them to easily setup and exploit mobile sensing experiments. Beyond the scientific contribution of this platform, the technical originality of ANTDROID lies in its Cloud orientation, whose implementation is based on an extension of the SCA component model that complies with the REST architectural style. This modular infrastructure provides a scalable yet open environment, which can be configured and customized according to the scientist requirements. We validate this platform by sensing various activities of mobile participants using Android smartphones.

**Key-words:** Mobile Sensing,Android, Privacy, Component-based Software Engineering

**RESEARCH CENTRE**
**LILLE – NORD EUROPE**

Parc scientifique de la Haute-Borne
40 avenue Halley - Bât A - Park Plaza
59650 Villeneuve d'Ascq

# AntDroid : Une plateforme répartie pour la collecte de traces d'activités mobiles

**Résumé :**

Les communautés scientifiques ont souvent recours à la simulation dans le but de valider leurs théories. Cependant, la pertinence des résultats obtenus est fortement connexe avec la qualité des données générées par les simulateurs. Ce phénomène est particulièrement vrai lorsque l'on considère les traces de mobilité humaine qui sont difficilement prévisibles. C'est dans ce contexte que nous introduisons AntDroid, une plateforme distribuée permettant de collecter les traces d'activités des utilisateurs de téléphone mobile. AntDroid offre aux communautés scientifiques une plateforme participative pour les aider à facilement mettre en place et à exploiter une expérience de collecte de traces. AntDroid est une plateforme évoluant dans les nuages, sont implémentation est basée sur le modèle de composant SCA, offrant ainsi une infrastructure modulaire et facilement configurable selon les besoins des scientifiques. Nous avons évalué notre approche en collectant des traces d'activités variées sur des participants utilisant des téléphones mobiles Android.

**Mots-clés :** Collecte mobile,Android, Vie Privée, Programmation orientée composant

# 1 Introduction

For years, the analysis of activity traces has contributed to better understand crowd behaviors and habits [31, 37]. For example, the *Urban Mobs* initiative [47] visualizes SMS or call activities in a city upon the occurrence of major public events. These activity traces are typically generated from GSM traces collected by the cellphone providers [55]. However, access to these GSM traces is often subject to constraining agreements with the mobile network operators, which prevents their free publication, while their scope remains limited to telecom data. Similarly, activity datasets are also exploited by the industry for billing customers or optimizing the delivery of products. For example, activity traces of taxi drivers have been thoroughly analyzed to understand the strategy adopted by a driver to maximize its income [37], but once again the company often restricts the access to the dataset.

In addition to that, activity traces are also used as a critical input for assessing the quality of scientific models and algorithms. As an example, the *Reality Mining* activity traces [41] collected by the MIT Media Lab or the *Stanford University Mobile Activity TRAces* (SUMATRA) [56] have become a reference testbed for testing mobile algorithms in ad hoc settings [49, 2]. The *Community Resource for Archiving Wireless Data At Dartmouth* (CRAWDAD) [32] is another initiative from the Dartmouth College aiming at building an archive of wireless-network traces. Nonetheless, the diversity of the activity traces available in these repositories remains limited and therefore constrains scientists to tune inadequate traces by mapping some of the parameters to their requirements. As an example, the activity traces of San Francisco taxi cabs are heavily used by the scientific community, but this dataset usually require to be downscaled in order to fit the requirements of a given mobile algorithm.

In this context, cellphones represent a great opportunity to collect a wide range of crowd activity traces. Largely adopted by population, with more than 290 millions sold in 2010 and an expected increase of more than 50 percents in 2011 according to Gartner institute, smartphone have become a central piece in people's living. Not only focusing on computing or communication capabilities, recent mobile devices are now equipped of a wide range of sensors enabling scientist to build a new class of datasets. For example, GPS receivers provide location information, gyroscope or accelerometer infer contextual information (*e.g.*, the user is sitting or walking [33]), while WiFi or bluetooth neighboring can be used to scan network access points as well as mobile devices in the vicinity. Furthermore, the generalization of *app stores* or *markets* on many mobile phone platforms leverages the enrollment of participants to a larger scale than it was possible previously.

Using cellphones to collect user activity traces is known in the literature either as *participatory sensing* [6], which requires explicit user actions to share sensors data, or as *opportunist sensing* where the mobile sensing application collect and share data without user involvement. These approaches have been largely used in the multiples research studies including traffic and road monitoring [44, 4], social networking [39, 40], environmental monitoring [45] or personal health monitoring [46, 8]. However, developing a sensing application allowing to collect a specific dataset over population in not trivial. Indeed, a participatory and opportunistic sensing application needs to cope with a set of key challenges [9, 34], including energy limitation, privacy concern, and requiring an deep expertise of mobiles devices. These constraints are making difficult, for scientists non expert in this field, to collect dataset for their studies. But more importantly, the developed ad hoc applications may neglect privacy and security concerns, resulting in the disclosure of sensible information. With the regard of the state of this art

in this field, we observe that current solutions lack of reusable approaches for collecting and exploiting crowd activity traces, which are usually difficult to setup and tied to specific data representations and device configurations. We therefore believe that mobile sensing platforms require to evolve in order to become more open and widely accessible to scientific communities. In this context, we introduce ANTDROID, an open platform targeting multiple research communities, which provides a lightweight way to develop and deploy an opportunistic sensing applications requiring a dedicated dataset while taking into account key challenges.

The reminder of this papers is organized as follows. We first identify the key challenges related to the development of mobile sensing platforms (cf. Section 2), before reporting on the design of the ANTDROID platform (cf. Section 3). Our proposal is assessed by evaluating the energy overhead and the scalability of the current implementation (cf. Section 4) and describing use cases involving this platform (cf. Section 5). Finally, we discuss extensively the related works in this domain (cf. Section 6) before concluding (cf. Section 7).

# 2 Mobile Sensing Challenges

The development of mobile sensing platforms is a sensitive and critical task, which requires to take into account a variety of requirements covering both technical and ethical issues [9, 34]. In this section, we therefore describe the key requirements that we considered as critical in this respect as well as the challenges induced for the ANTDROID platform.

## 2.1 Ethical Challenges

**Privacy management** is one of the most critical requirements to be covered by mobile sensing platforms. Several studies [18, 17] reveal that geolocated datasets (even when they are anonymized) can be attacked to extract common places of participants (*e.g.*, work/home addresses) and to predict with a relatively good accuracy the next location of an individual as well as his current activities [35]. Preserving the user privacy must therefore be a fundamental concern of any mobile sensing platform. As more and more people are becoming sensitive to the disclosure of their personal information, mobile sensing platforms should provide enough confidence in the respect of their privacy.

**Sensing control** is another issue that requires a careful consideration and refers to the control given to the user to enable and discard the sensing of specific information. While the scientist can be interested in sensing informations, such as neighboring devices (using Bluetooth), the user might rather prefer not to sense this information for privacy or energy reasons. The mobile sensing platform should therefore provide control preferences to select the accuracy and the availability of sensors

**User acceptation** covers the definition of appropriate levers to involve users in the mobile sensing platform. As the objective of a mobile sensing platform is to collect realistic datasets, it is important that the user contributes positively to the mobile sensing activities. As described in [9], recruiting participants is difficult and mobile sensing platforms should rather invest on the definition of rewarding mechanisms to encourage users to contribute to the sensing tasks.

## 2.2 Technical Challenges

**Energy consumption** is a key technical challenge with regards to the resulting solution proposed to the mobile user. In particular, the mobile sensing platform should not drain the battery of the mobile sensing when collecting activity traces. Although mobile devices tend to be very powerful computing devices, they are still sensitive to power-aggressive sensors, such as GPS or Bluetooth, continuously stressed by applications.

**Experiment deployment** deals with the deployment of the mobile sensing activities on mobile devices. Although *app stores* offer a good way to good way to disseminate a sensing experiments to a large population, the main problem the lose the control of the running experiment. If the first data collected are not satisfying, it becomes hard to change the behaviour of the sensing application once deployed in the wild. Moreover, deploying via an *app store* an application raises the problem of dealing with potentially massive amount of data irrelevant to the scientist. Current *app stores* do not provide mechanism for limited to download an application to a subset of users depending of their characteristics (*e.g.*, location, type of devices, type of sensors). We want to address this issue by integrating a dedicated *app store* for sensing applications offering the possibility to a scientist requirements of the deployment of the tracking application and change the behaviour of the application once deployed.

**Platform customization** refers to the diversity of technologies, which can be used to implement the mobile sensing platforms. This diversity covers both the type of mobile devices used to collect activity traces and the software solutions used to store and process the collected dataset. Therefore, this challenge requires to offer an open platform, which can be easily tailored to the requirements of scientists depending on the nature of the activity traces they are interested in collecting.

**Summary** With regards to the challenges identified by the state-of-the-art, we believe that mobile sensing platforms have to meet both ethical and technological challenges. While some of these challenges requires a careful attention and might be particularly difficult to fulfill, we believe that an open platform approach can help to leverage mobile sensing practices and help in fostering the validation of research theories by supporting the sensing of realistic dataset from the crowd. To succeed, such a mobile sensing platform requires to focus on the end-user in order to provide her/him enough guarantees with regards to privacy, usability and energy consumption. From a scientific perspective, this platform should provide dedicated tools and infrastructures to leverage the definition and the continuous deployment of sensing tasks on mobile devices. These challenges are the foundations of the ANTDROID platform and the following sections report on the solutions that have been developed to provide an open platform to ease the sensing of activity traces.

# 3 The AntDroid Platform

The ANTDROID platform distinguishes between two roles. The former, called *scientist*, is a researcher who wants to define and deploy an experiment over a large population mobile users. The platform therefore provides a set of services allowing her/him

to describe experiment requirements in a domain-specific language, deploying experiment scripts over a subset of participants and connect other services to the platform to extract and reuse dataset collected in other context (*e.g.*, *visualization*, *analysis*, *replay*). The scientist is offered a web environment to define, deploy, and store the collected dataset. In particular, ANTDROID is built on the principles of Cloud computing and offers a modular service-oriented architecture, which can be customized upon scientist requirements. The latter is the mobile phone user, also called *participant*. The ANTDROID platform provides a mobile application allowing to download experiments, running them in a dedicated sandbox and uploading datasets collected to the ANTDROID server. The mobile application provide several mechanisms in order to control user privacy and battery lifespan.

To present the ANTDROID platform, we use a sample scenario, which consists in building a map of GSM signals strength in a specific geographic area (*e.g.*, the faculty campus) from user activity traces. We chose this scenario as it illustrates that ANTDROID does not restrict itself to the sensing of mobility traces, but can also be used to observe other contextual information like the signal strength of GSM antennas. To define this map, the scientist uses the ANTDROID platform to define its sensing experiment and to collect the appropriate activity traces from participants. In this section, we therefore first report the server side and then the client side of the ANTDROID platform before assessing the challenges we discussed in the previous section.

## 3.1 The Scientist Web Infrastructure

The main objective of ANTDROID is to provide to scientist a platform, which is open, easily extensible and configurable in order to be reused in various contexts. To achieve this goal, we design the server-side infrastructure of ANTDROID as an SCA application (cf. Figure 1). The *Service Component Architecture* (SCA) [1] is a set of specifications for building distributed application based on *Service-Oriented Architectures* (SOA) and *Component-Based Software Engineering* (CBSE) principles. SCA specifies a hierarchical component model, which means that components can be implemented either by primitive language entities or by subcomponents. In the latter case the components are called *composites*. Each component define *services* (or provided interface), and *references* (or required interfaces). The references and services are connected by means of *wires*. SCA is designed to be independent from programming languages, *Interface Definition Languages* (IDL), communication protocols, and non-functional properties. In particular, to support interaction via different communication protocols, SCA provides the notion of *binding*. For SCA references, bindings describe the access mechanism used to invoke a service. In the case of services, the bindings describe the access mechanism that clients use to execute the service. We therefore believe that SCA provides an flexible foundation for the ANTDROID infrastructure by accommodating a wide diversity of programming languages and communication protocols.

All components building the server-side infrastructure of ANTDROID are hosted by a cloud infrastructure. The *Scientist* and *Paticipants* components are the entries points for user involving in the platform. Both components defined all the services that can be invoked remotely by scientists or participants. Remote services are exposes as a REST resources. Additionally to the components hosted in the cloud, scientists can easily develop and deploy their own components in order to tune the platform according to their requirements. By supporting various implementation languages, such as Scala, BPEL, Scripting languages, XQuery, OSGi, Spring, scientists are not constrained to use a specific language for implementing their components. We present an example in

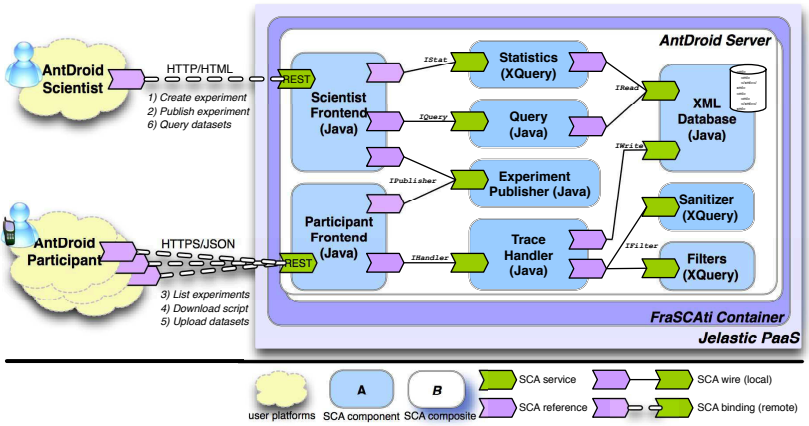



Figure 1: Architecture of the ANTDROID Software Platform.

Section 3.1.5.

In order to ease the adoption of the ANTDROID platform and manage services provided by the platform, we developed a Web interface. Thus, this choice does not require the scientist to install any specific software on her/his computer and it leverages the burden of handling network-level configurations in order to deploy and maintain a server infrastructure to store the data collected by the participants. Therefore, ANTDROID is available as a website[1] to which any scientist can connect to. Once authenticated, the scientist can create new experiments, follow their progression, and exploit the collected data directly from this interface.

### 3.1.1 AntDroid Scripting Language

To reduce the *learning curve*, we decided to adopt a standard scripting language to ease the description of experiments by the scientist. We therefore propose the ANTDROID *Scripting Language* as an extension of JavaScript, which provides an efficient mean to describe an experiment without a specific knowledge of mobile device technologies where the scripting language will be executed. The choice of JavaScript was mainly motivated by its native support for JSON (*JavaScript Object Notation*), which is a lightweight data-interchange format reducing the communication overhead. Furthermore, processing JSON documents in Android is slightly more efficient than XML [52] and can be easily to be transformed in XML format whenever needed. For example, Listing 1 describes the piece of script to be written by the scientist to collect geolocated GSM signal strength traces. This script is triggered whenever the position of the participant changes and builds a new trace out of the sensed data.

[1] ANTDROID: `http://antdroid.jelastic.com/web`

```
1   var experiment = new Experiment("GSM_Signal_Strength")

3   // event triggered when location changes
4   experiment.onLocationStateChange(event) {
5   return { "trace": {
6           "lat": event.getLatitude(),
7           "lon": event.getLatitude(),
8           "time": event.getTime(),
9           "ss": event.getSignalStrengthLevel()
10  }}}

12  // start collecting activity traces
13  experiment.start()
```

Listing 1: WiFi Networks Sensing Example

The first step consists in creating a new experiment (line 1). Each experiment is associated with a local database used to store the produced traces in the mobile device. Each trace collected by the experiment is stored in the database before to be aggregated and sent to AntDroid server infrastructure. In this experiment, the script requires to access two sensors: the *Location sensor* to collect periodic updates of the device's geographical location, while the *WiFi sensor* monitors the different WiFi networks detected. The script therefore requires to request access to these sensors if the user permissions grant it (lines 3–4).

When the state of an active sensor is updated, the system creates an event and triggers the handlers associated to this event. In our example, we are interested in collecting location and Wifi Network only when the location of the user changes. Therefore, the scientist only needs to implement the function `onLocationStateChange` as shown in Listing 1 (line 6). The function can access all raw sensors data via the object `event` passed as a parameter.

The last step consists in describing the content of the activity trace. The AntDroid *Scripting Language* provides several *helpers* in order to build traces for geolocation standard formats, such as *GPS Exchange Format* (GPX) or *Keyhole Markup Language* (KML). In addition to these helpers, the scientist is free to define a custom representation and return an object directly in JSON representation (lines 7–11). The last line of the script is used to register the script within the AntDroid engine and to start collecting data.

In addition to the sensors reported in this example, the AntDroid Scripting Language supports the monitoring of phone calls, Wifi networks, application status (installed, running), networking, short messaging, system state and can be easily extended.

**Privacy Filters.** In addition to this script, the scientist can define privacy filters to limit the volume of collected data and enforce the privacy of the participants. In particular, AntDroid currently supports two types of filters:

**Area filter** allows the scientific to specify a geographic area where the data requires to be collected. In our example, this area maps to the place where the scientist is interested in collecting the WiFi networks (*e.g.*, campus area). This filter guarantees the participants that no data is collected and sent to the AntDroid server outside of this area.

**Period filter** allows the scientific to define a time period during which the experiment should be active and collect data. In our example, this period can be specified during the office hours in order to discard data collected during night, while the participant is expected to be at home.

By combining these filters, the scientist preserves the privacy of participants, reduces the volume of collected data, and improves the energy efficiency of the mobile application (cf. Section 3.2).

### 3.1.2 AntDroid Deployment Model

Once an experiment is defined using the scripting language, scientists can publish it into the *Experiment Publisher* component in order to make it available to participants. Once published, two deployment strategies can be considered for deploying experiments. The former, called pull-based approach, is a proactive deployment strategy where participants download the list of experiments from the remote server. The latter, push-based approach, propagates the experiments list updates to the mobiles devices of participants. In the case of ANTDROID, the push-based strategy would induce a communication and energy overhead and, in order to leave the choice to participants to select the experiments they are interested in, we adopted the pull-based approach as a deployment strategy. Therefore, when the mobile device of a participant connects to the *Experiment Publisher*, it sends its characteristics (including hardware, location, sensor available and sensors that participants want to share) and receives the list of experiments that are currently published. The scientists can configure the *Experiment Publisher* to limit the visibility of their experiments according the characteristics of participants. In order to reduce the privacy risk, device characteristics sent by participants are not stored by the infrastructure and scientist cannot access to this information.

Additionally, the *Experiment Publisher* component is also used to update the behaviour of the experiment once deployed in the wild. When an opportunistic connection is established between the mobile device and the ANTDROID server, for example when a report of collected data are submitted, the version of the experiment deployed in the mobile device is compared to the last version published in the server. The experiment is eventually updated when the latest version of the experiment without constraining participants to re-download manually the experiment. In order to avoid any versioning problem, each uploaded data trace includes a key encoding the version of the experiment used to collect data. Thus, scientists can configure the *Experiment Publisher* component in order to keep or discard data collected by older versions of the experiment.

### 3.1.3 AntDroid Trace Dashboard

As discussed in Section 2, one main challenge of our platform is to support multiple data representations according to scientist requirements. Current platforms [16] use traditional SQL database to ensure data persistences in their back-end server. However, this design choice does no provide enough flexibility to the scientist, while it raises several privacy issues. Indeed, a majority of platforms define static dataset schemas including all the possible information that can be collected from a mobile device. By doing so, the privacy of participants is compromised, while the amount of uploaded data does not scale. In ANTDROID, we rather advocate the definition of targeted experiments where only specific data is collected by scientists. This means that *i)* ANTDROID participants are clearly aware of data collected by experiments, *ii)* data can be processed in the device to sanitize and reduce the collected, and therefore *iii)* the amount of uploaded data traces is reduced to the meaningful set of information requested by the scientist. As a side effect, this choice requires the definition of a flexible storage mechanism to accommodate the diversity of datasets collected by scientists.

For example, geospacial data can be represented as a couple of decimal values (longitude/latitude) for a fined-grained representation or generalized (*e.g.*, area, city, state) for more coarse-grained representations. In that case, using SQL database requires the scientist to define a specific database schema, which does not help in leveraging the definition of experiments. We therefore chose to use a native *eXtended Markup Language* (XML) database to store the collected datasets in the ANTDROID server. The major benefit of this technology prevents us to impose a specific schema for storing datasets. Moreover, recent advances in XML databases have demonstrated the scalability of this technologies [23, 27]. As illustrated in Listing 1, the data schema is automatically inferred from the JSON object produced by the script. Once aggregated and sent to the ANTDROID server, JSON objects are transformed into XML documents, filtered by the *Trace Filter* component and then stored in the XML database. *Scientist* can then process the collected data, by executing XQuery scripts [5] from the web interface or by deploying a specific component bound to the service exposed by *XML Database* component as depicted in Figure 1.

#### 3.1.4 AntDroid Participants Involvement

In order to attract participants, a sensing platform has to provide appropriate levers to catalyze the collection of datasets. As cited by [11], one key challenge when cellphone are used as research platforms is to incite people to participate to a given experiment. Even if sensing applications represent a great interest for scientists, it does not offer any particular service to the participants, while consuming their resources (*e.g.*, battery, bandwidth). To help scientist to motivate participants to contribute to their experiments, we provide a rewarding mechanism in order to encourage participants to collect relevant datasets.

As the ANTDROID mobile application provides several mechanisms to control the access to sensors, the rewarding mechanism is based on the quality and the volume of datasets produced by participants. Therefore, the more sensors are activated by participants and the more datasets are uploaded to the ANTDROID servers, the more credits the participant receives for its involvement in the experiment. The credits can be configured by the scientist in order to privilege the retrieval of specific data. For example, the scientist can allocate more credits to the GPS sensors in order to balance to the energy consumption and the privacy sensitive of this sensor. The assigned credits are then used by the scientist to provide participant rankings, involvement badges, or even coupons to reward the users.

#### 3.1.5 AntDroid Platform Extensions

As already reported in this section, the server-side architecture of ANTDROID is built as an SCA application, thus benefiting from the modularity of SCA standard, which allow to easily extends the platform. An SCA component description mainly defines component implementation languages as well as communication protocols that are required to deploy the component. Listing 2 describes a specific component to be deployed by the scientist in the server infrastructure. In this example, the component access the content of the XML database to build a KML document, which can be downloaded and displayed with Google Earth. This component is implemented in Java (line 6) and is exposed as a REST resource (lines 8–11). In order to retrieve data stored in the XML database, the component declares a reference with a SCA binding to access the query service exposed by the *Query* component (lines 13–16).

```
1       <composite name="antdroid-kml-extension"
2               xmlns="http://www.osoa.org/xmlns/sca/1.0"
3               xmlns:frascati="http://frascati.ow2.org/xmlns/sca/1.1">

5               <component name="antdroid-kml-extractor">
6                       <implementation.java class="antdroid.KMLExtractor" />
7                       <service name="export">
8                               <interface.java interface="antdroid.IExportKml"/>
9                               <frascati:binding.rest uri="/kml-document"/>
10                      </service>
11                      <reference name="xquery-engine">
12                              <interface.java interface="antdroid.api.IQuery"/>
13                              <binding.sca/>
14                      </reference>
15              </component>

17      </composite>
```

Listing 2: KML Extraction Component

This short example demonstrates how scientists can extend the ANTDROID server-side platform to benefit from the resources of the remote infrastructure to process the potentially huge amount of collected data. Additionally, we already provide off-the-shelf components which can be used to visualize and replay data collected by experiments. There components are illustrated in Sections 4 and 5.

## 3.2 The Participant Mobile Application

Although our solution could be extended to other *Operating Systems*, the ANTDROID mobile application is currently based on the Android operating system for the following reasons. First, the Android operating system is popular and largely adopted by population, unit sales for Android OS smartphones ranked first among all smartphone OS handsets sold in the U.S. during the first half of 2010. Secondly, Android is an open platform supporting all the requirements for continuous sensing applications (*e.g.*, multitasking, background processing and ability to develop an application with continuous access to all the sensors), while for example, iOS 4 does no permit a continuous accelerometer sampling.

A participant willing to be involved in an experiment proposed by a scientist can download the ANTDROID mobile application by flashing the QR code published on ANTDROID website, install it, and create an account on the remote server infrastructure. Once registered, the HTTP communications between the mobile device of the participant and the remote server infrastructure are authenticated and encrypted in order to reduce potential privacy leaks when transfer the collected datasets to the ANTDROID server.

### 3.2.1 AntDroid Phone Software

Figure 2 depicts the ANTDROID software architecture. Building on the top of Android SDK, this architecture is mainly composed of four main parts allowing *i*) to interpret experiment scripts (*Facades*, *Scripting engine* and *AntDroid Scripting API*) *ii*) to establish connection with the remote server infrastructure (*Server Manager*), *iii*) to control the privacy parameters of the user (*Privacy Manager*), and *iv*) to control power saving strategies (*Battery Manager*).

**Scripting Engine.** *Sensor Faces* bridge the Android SDK with the*Scripting Engine* provided by the Android Scripting project [22]. By using JSON-RPC communications, this mechanism an easy way to implement Java function and thus allow to be called

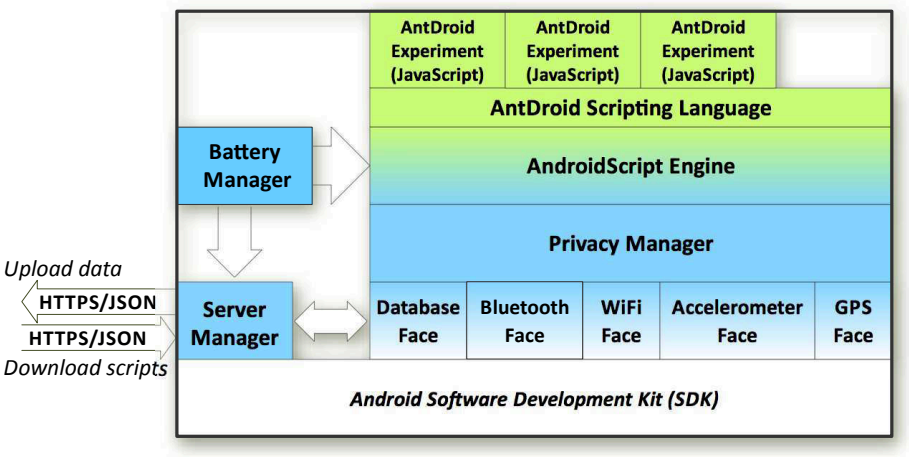



Figure 2: Architecture of the ANTDROID Software Architecture

directly from script. We build an abstract layer *AntDroid Scripting*, which implement the sensors accessed by the experiment scripts. This abstract layer covers two roles: a *security role* to prevent malicious calls of critical code and a *accessibility role* to leverage the development of sensing experiments as presented in Section 3.1.1.

**Battery Manager.** Although the last generation of smartphone provides very powerful computing capabilities, the major obstacle to realize continuous sensing application refers to their energy restrictions. Therefore, in order to reduce the communication overhead with the remote server, which tends to be energy consuming [52], datasets are uploaded only when the mobile phone is plugged. In particular, the battery manager component monitors the battery state and triggers the server manager component when the battery starts charging in order to send all the collected datasets in the remote database. Additionally, this component monitors the current battery level and suspends the scripting engine component when the battery level goes below a specific threshold (20% by default) in order to stop all running experiments.

**Privacy Manager.** In order to cope with the *ethical challenges* we raised, the ANTDROID mobile application allows *participants* to define privacy rules limiting experiments to collect data depending on their preferences. As depicted in Figure 3, three categories of privacy rules can be defined: *i) location rules* and *ii) time rules* specify geographical zone or time intervals conditions under which experiments are authorized to collect data. All the privacy rules defined by the participant are interpreted by the *Privacy Manager* component, which suspends the scripting engine component if one this rules is triggered. The last category of privacy rules refer to *iii) authorization rules*, which prevent sensors activation or access to raw sensor data if the user does want to share this information. Additionally, a processing mechanism use cryptography hashing to prevent experiment to collect sensitive data, such as phone numbers, SMS text, or address book.

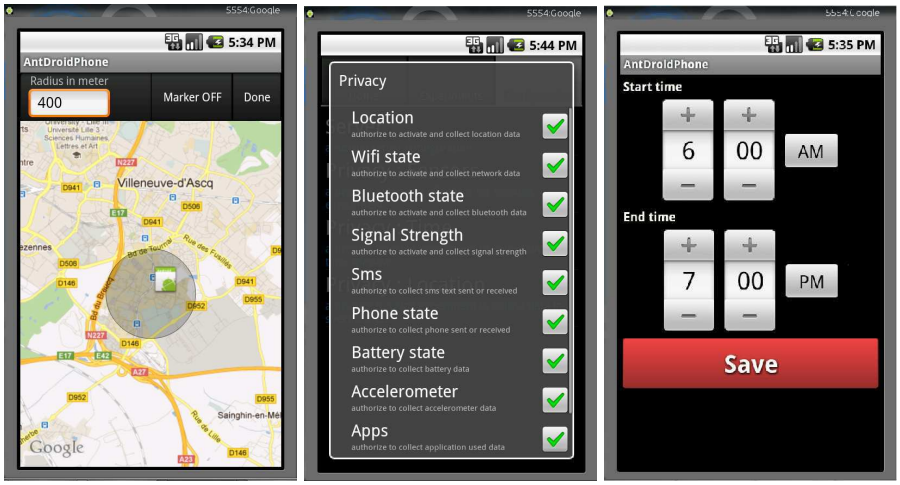



Figure 3: Privacy rules configuration

# 4 Evaluation

This section reports on different experiments we conducted in order to evaluate our solution. The server infrastructure uses the *FraSCAti* [51] middleware dedicated to the development of SCA applications. For the XML database, we using BaseX [24, 27] including XPath/XQuery processor and full support for the W3C Update and full text extensions. The resulting platform is hosted on the *Jelastic* cloud infrastructure, using a free account and configured with Java 1.6 and one instance of Apache Tomcat 6.0.

## 4.1 Building GSM Signal Strength Maps

In this use case, we have deployed the sensing experiment presented all along of this paper. The experiment, described in Listing 1 has been developed and deployed using the Web interface provided by the ANTDROID server infrastructure (cf. Figure 4). The objective of this experiment is to build a map displaying the GSM network coverage of our campus for a specific network operator. The experiment records the GSM signal strength and the location of the participant when sensing this information. This information is only collected when the participant moves more than 50 meters away from his previous collecting position with a maximum interval of 1 minute, which is default configuration provided. As we are only interested in building the map of a specific area, we specify a privacy location filter in order to collect relevant dataset and to limit the deployment of experiments to participants located in a radius of 1000 meters around of the campus. To focus on a specific mobile operator, we configure the *Experiment Publisher* in order to authorize only mobile phones with our specific operator required to download this experiment. This experiment has been downloaded by four participants who were compatible with the experimental conditions (location and phone operator). Figure 5 depicts the resulting dataset collected during one week. On average, each participant has produced 80 traces per day, which represents a memory footprint of $11.8\ kB$. Both figures were generated by the visualization component provided by ANTDROID platform. The first figure is a Google Earth representation of the collected data of the campus in three dimension where the height of the polygons corresponds to

of GSM signal strength level. The second figure displays the same collected dataset on Google Map, and uses the shade of color to map the GSM signal strength level. The darker, the higher the signal strength is.

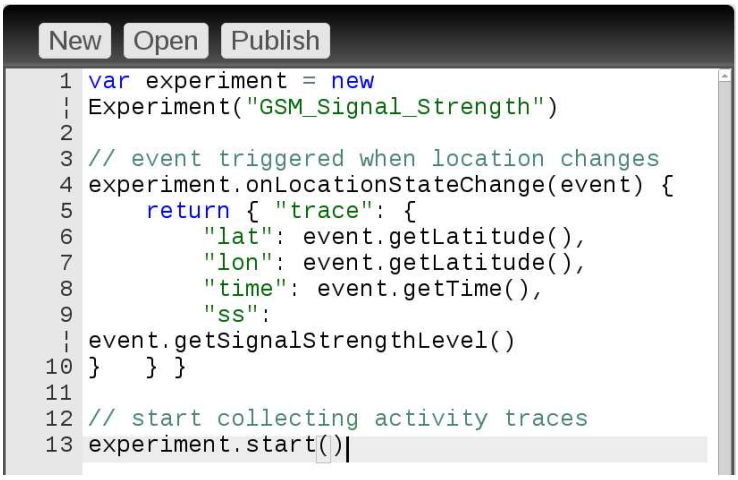



Figure 4: Screenshot of the scientist web interface.

## 4.2 Evaluating the Energy Consumption

Participating to an experiment should have a minimal impact on the battery consumption the mobile phone for not interfering with its normal usage. This second evaluation aims at evaluating the battery lifespan impact of the AntDroid mobile application. As this cost strongly depends on *i)* the nature of the experiment, *ii)* the types of sensors accessed, and *iii)* the volume of produced data, we conducted a first experiment in order to identify the impact of running an experiment collecting a minimal set of information. In order to limit minimize the noise produced by other applications, we turned off Bluetooth and Wifi interfaces and we stopped all the applications and services running in background. The curve labelled AntDroid in Figure 6 reports on the result of this experiment, which has been executed on a Samsung Galaxy S based on

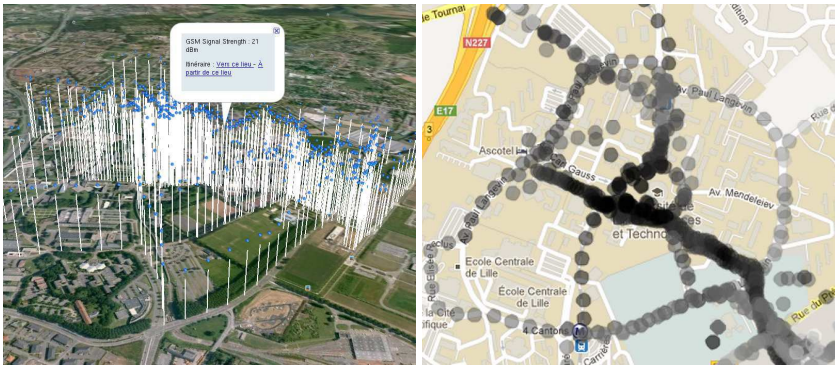



Figure 5: GSM Signal Strength Experiment Visualizations.

Android 2.2. As we can see, the baseline experiment tends to have a very small impact on the battery lifespan, thus highlighting the low overhead induced by the ANTDROID mobile application. Then, we initiated a second experiment in order to evaluate the impact of energy-consuming sensors that can be used to collect data (cf. Figure 6). For this experiment, we developed three additional scripts, which we deployed separately. The first script, labelled AntDroid + Bluetooth, triggers a bluetooth scan every minute and collect both the battery level as well as the resulting bluetooth scan. The second script, AntDroid + GPS, records every minute the current location collected from the GPS sensor, while the third script, AntDroid + WiFi, collects a WiFi scan every minute. These three experiments demonstrates that, even when stressing energy-consuming sensors, it is still possible to collect data during a normal day of work without recharging the mobile phone.

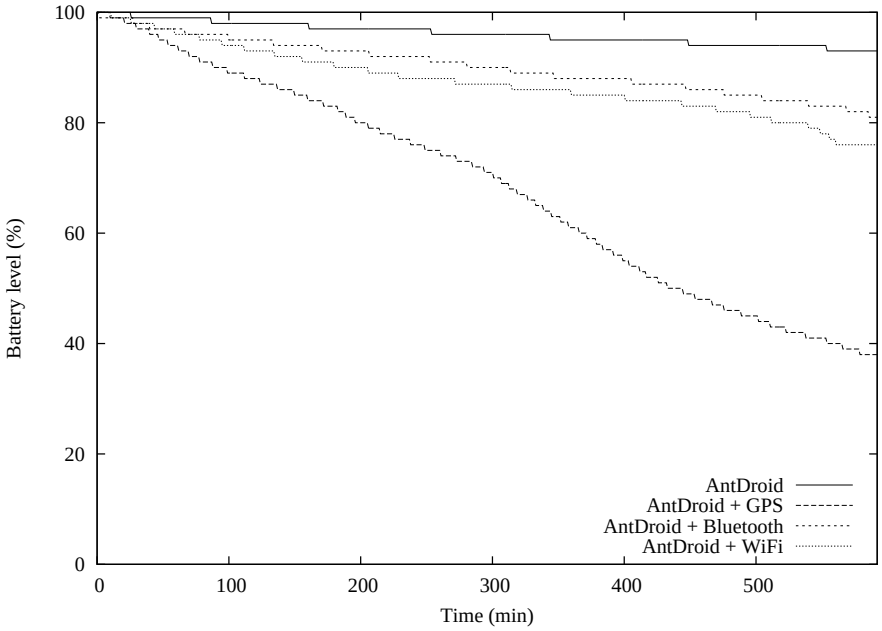



Figure 6: Impact of ANTDROID on the battery lifespan.

As these scripts are not optimized, the observed energy consumption can easily be reduced by adapting the triggers and using local processing. As expected, the GPS sensor tends to consume much more energy than the other sensors. Thus, based on these observations, we plan to investigate the definition of an energy model for ANTDROID scripts in order to provide an energy feedback to participants prior to the deployment of experiments.

## 4.3 Evaluating the Server Scalability

The second evaluation aims at demonstrating the robustness and the scalability of our server infrastructure with two scenarios. The first scenario consists in simulating dataset upload bursts corresponding to the concurrent transfer of datasets initiated by 1 to 100 participants. Each dataset has a size of $73.4\ kB$ (500 traces) and the displayed curve reports on the mean latency observed for transferring 100 datasets per

participants (cf. Figure 7). As expected, uploading datasets from a mobile device tends to take time, however ANTDROID triggers this operation when connecting the power cord to the mobile device (*i.e.*, once a day), which means that this expansive process does not impact the battery lifespan of the mobile device. Furthermore, the critical situation where $100$ participants would plug simultaneously their mobile device to the power cord is considered as seldom, and we rather observed that the mean latency for uploading datasets in normal deployments is around $6\ seconds$. Such a latency is considered as almost imperceptible from a participant perspective.

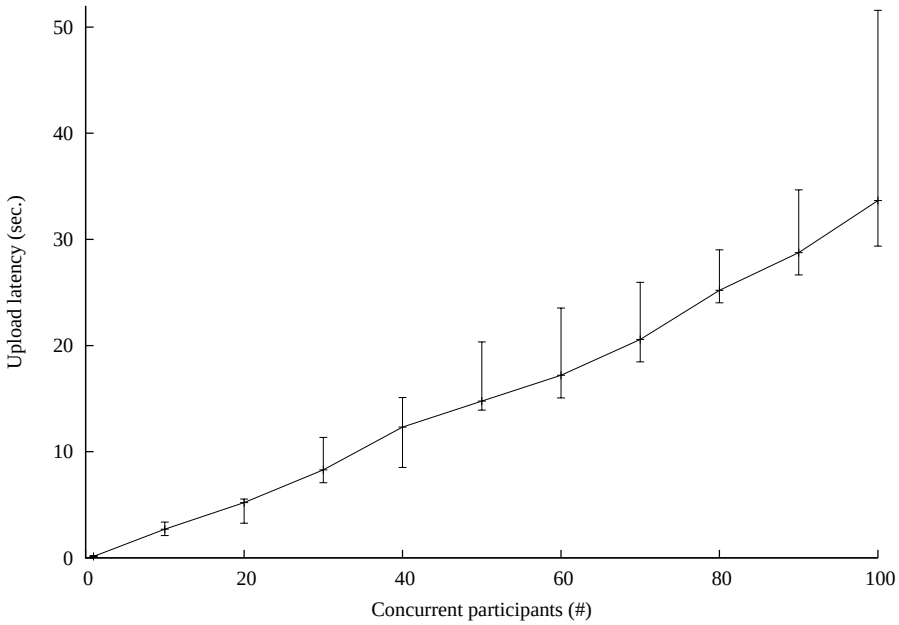



Figure 7: Impact of concurrent uploads on ANTDROID.

The second scenario consists in processing a dataset containing up to 3 Million of traces collected by the ANTDROID platform. In particular, we considered three queries consisting in counting all the traces available in the database, counting the traces collected during a specific period, and finally counting the traces uploaded by a given participant. These scripts are developed as XQuery scripts, which are processed against the XML database by executing them from the ANTDROID web environment. Figure 8 therefore reports on the mean time spent in processing each of these requests against datasets exhibiting different sizes. One can observe that constraining the query space improves the processing latency although the processing of a dataset made of 3 Million of traces can still be completed within $10seconds$.

# 5 Use Case Perspectives

This section presents additional use cases that are currently covered by the ANTDROID platform. By going beyond classical mobile sensing experiments, the use cases reported in this section demonstrate the applicability of ANTDROID to a wide diversity of requirements.

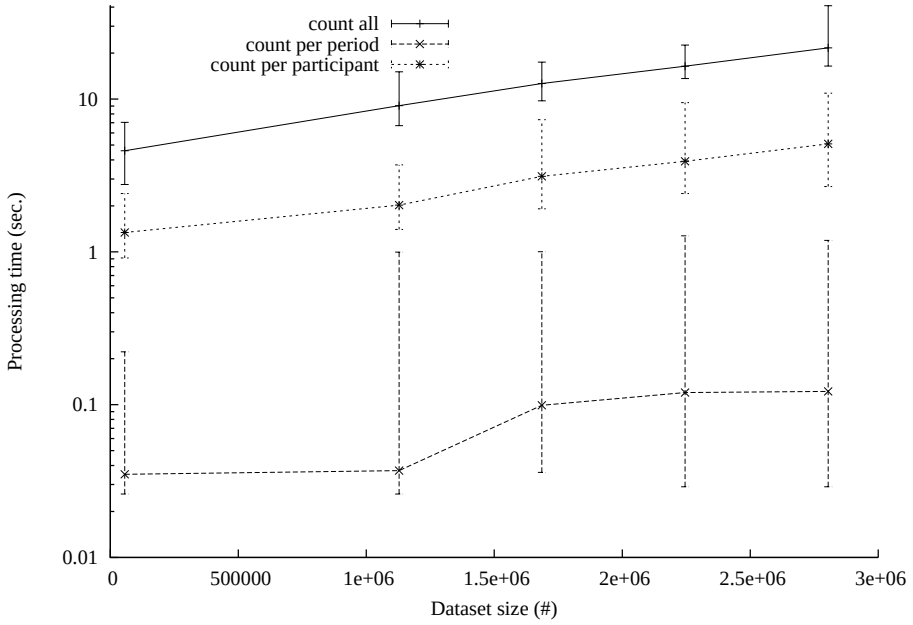



Figure 8: Dataset processing performances in ANTDROID.

## 5.1 Testing Complex Event Processing Systems

This use case demonstrates how our platform can contribute to improve testing of location-based applications, such as traffic monitoring systems [29]. Such systems usually need to cope with events continuously generated by sensors. To monitor and react in real-time to these events, *Complex Event Processing* (CEP) [38] has been proposed has a scalable paradigm to build these systems [14]. A CEP correlates events by identifying *event patterns* from *event streams*. One of the key features of CEP is the definition of reactive rules matching *event patterns* to trigger specific actions.

However, partly due to the unpredictability of continuous arriving events, it is difficult for the developer to test the developed rules prior to their deployment within the CEP. We therefore believe that the ANTDROID platform can help developers to test this kind system under more realistic conditions. In particular, we use ANTDROID to inject events, which have been previously collected by the platform. To support such realistic testing scenarios, we first connect the CEP to the server infrastructure of ANTDROID by deploying a specific rules within the CEP engine to forward the incoming events to the ANTDROID platform. This step allows the developer to build a realistic dataset containing events that are highly relevant for her/his application. As the ANTDROID server infrastructure builds on standard web protocols, the definition of this forwarding rule can easily be achieved by using remote connectors of existing CEP engines. Once collected, the datasets (or a specific subset of it) can be replayed by the ANTDROID platform to simulate realistic execution conditions of the CEP. The developer can therefore deploy the set of rules she/he want to test and use specific assertion rules provided by the framework to check that the system behaves as expected.

Furthermore, we intend to apply the same approach to validate mobile ad-hoc algorithms [59, 13] by configuring state-of-the-art simulators with such realistic datasets.

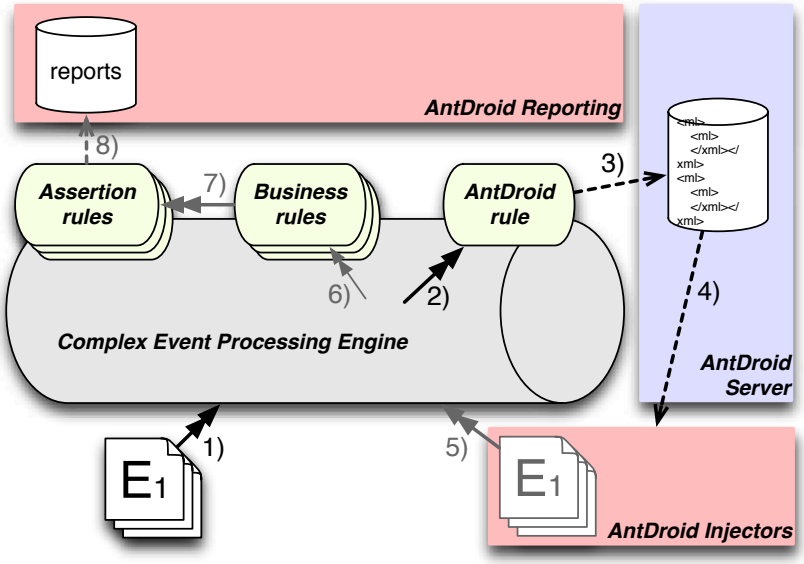



Figure 9: Realistic testing phases

### 5.2 Building Mobility Models

This use case contributes to the continuous improvement of privacy support within the ANTDROID. This study therefore consists in checking the privacy level of ANTDROID experiments by executing privacy attacks against the collected dataset to observe if sensitive information can be inferred from sanitize datasets. In particular, we collaborate with a team working on the inference of mobility models from mobility datasets [30, 19, 18, 17]. This type of attacks demonstrates the sensitive nature of geolocation data and show that by building user-centered mobility models, it is possible to guess the identity of the associated participant. However, ANTDROID does not specifically focus on geolocated activity traces and offers enough control to the participants to disable the access to the GPS sensor.

More recently, the UIC Shuttle Scheduling Service [4] has demonstrated that the inference of mobility models from geolocation information can contribute to provide better services to the end-users. This service periodically collects the position of buses to provide an up-to-date map of bus lines and and estimation of the expected arrival of the next bus depending on your position. Although the UIC Shuttle Scheduling Service does not use ANTDROID platform currently, we believe that ANTDROID can contribute to the development of a new generation of advanced services exploiting contextual information collected by users to provide better services.

## 6 Related Work

**Activity Traces.** As mentioned earlier in this paper, a plethora of activity traces are already available on the Internet [32, 56, 57]. However, these activity traces are predefined datasets, which may be inappropriate for helping the scientist in validating their models and algorithms. For example, most of the CRAWDAD datasets [32] were produced in a limited vicinity, either during a conference, or on a campus. In addition to

that, a minority of the available datasets includes accurate localization data, while most of them focused on contact traces. Phone operators are another potential source of information for collecting activity traces of mobile users. Unfortunately, phone operators are not keen to release their datasets and do so to only some lucky few [55, 47].

**Tracking Platforms.** There are very few data collection tools that are freely available on the market. For example, the software used to build the Reality Mining dataset [41] is not maintained anymore [48]. Data collection tools has attract most interest in the literature. SYSTEMSENS [15], a system based on Android, focuses on collecting usage context (*e.g.*, CPU, memory, network info, battery) of smartphones in order to better understand the battery consumption of installed applications. To minimize the battery impact, SYSTEMSENS has been designed to be unobtrusive by avoiding any user interface. Similarly, LIVELABS [53] is a tool proposed to measure wireless networks in the field with the principal objective to generate a complete network coverage map in order to help client to select network interface or network operation to identify blind spots in the network. However, such tools are closed platforms, designed for collecting specific datasets and cannot be reused in unforeseen contexts in contrast to ANTDROID. Furthermore, these projects are typical experiments that be deployed by platform, while providing privacy guarantees.

More interestingly, MYEXPERIENCE [16] is a system proposed for Windows mobile smartphones, tackling the *learning curve* issue by providing a lightweight configuration language based on XML in order to control the features of the application without writing C# code. MYEXPERIENCE collects data using a *participatory* approach—*i.e.*, by interacting with users when a specific event occur (*e.g.*, asking to report on the quality of the conversation after a call ends). However, MYEXPERIENCE does not consider severals critical issues, such as maintaining the privacy of participants or the strategic deployment of experiments. Even if an experiment can be modified in the wild, each experiment still requires a physical access to the mobile device in order to be installed, thus making it difficult to be applied on a large population of participants.

In the literature, severals deployment of sensing applications strategies have been studied. For example, ANONYSENSE [54] uses—as ANTDROID—a pull-based approach where mobile nodes periodically download all sensing tasks available on the sever. A sensing task is written in a dedicated language and defines when a mobile node should sense and under which conditions the report should be submitted to the server. However, ANONYSENSE does not provide any mechanism to filter the mobile nodes able to download sensing task, thus involving a communication overhead if the node does not match the sensing task downloaded. On the contrary, PRISM [10] adopts a push-based approach to distribute sensing tasks over mobile nodes. PRISM is a platform, running on Microsoft Windows Mobile 5.0, and supporting the execution of generic *binary code* in a secure way to develop real-time participatory sensing application without reinventing the wheel. To support real-time sensing, PRISM server needs to keep track of each mobile node and the report they periodically send (*e.g.*, current location, battery left) before selecting the appropriate mobile phones to push application binaries.

**Privacy & Security.** The sensitive nature of activity traces requires to carefully consider privacy and security issues. *GEoPrivacy-Enhancing TOolkit* (GEPETO) [18, 19] is a software toolkit providing researchers, who are concerned with geo-privacy, means to evaluate various sanitization techniques and inference attacks on geolocated data. This toolkit includes heuristics for identifying important places, so called *Points Of Interests* (POIs) or predicting the movement patterns of individuals by attacking some

geolocated data that is expected to be sanitized. ANTDROID provides a limited support to geo-privacy issues by including various types of *filters*, which can be used to filter out or sanitize sensitive information.

*AnonySense* is an opportunistic collection platform with a strong emphasis on privacy. By combining novel techniques, such as *k-Anonymity* and *tessellation*, reported data collection time and location are blurred in order to preserve the privacy of participants and thus prevent attacks on geospacial data. Despite this techniques has been largely adopted in multiple studies [3, 25], the cost of privacy enforcement implies the degradation of the quality of data, thus also potentially decreasing its utility. Even if this kinds of protection mechanisms can be sufficient in some case, it can be irrelevant for studies requiring a fine-grained location of participants.

*SensorSafe* [7] is an other participatory platform, which allows users to share data with privacy guaranties. As our platform, *SensorSafe* provides fine-grained temporal and location access control mechanisms to keep the control of data collected by sensors on mobile phone. However, participants have to define their privacy rules from a web interface while in ANTDROID these rules are defined directly from the mobile phone.

TAINTDROID [12] is a dynamic taint tracking and analysis system capable of simultaneously tracking multiple sources of sensitive data. TAINTDROID is built as an Android service, which monitors sensitive data and provide informed use of third-party applications for phone users and valuable input for smartphone security service firms seeking to identify misbehaving applications. In ANTDROID, users are volunteer participants and are thus informed *a priori* of the transmission of potentially sensitive data. Nonetheless, ANTDROID could be easily combined with TAINTDROID to notify users of the activity traces that are sent to the ANTDROID infrastructure.

**Location-aware Applications.** The *San Francisco Innovations Showcase* [50] is a directory listing mobile applications related to the city of San Francisco. Although this approach encourage the development of geo-tracking mobile applications, it does not offer a common infrastructure for collecting and exploiting the activity traces. Our platform rather proposes a reusable infrastructure for developing various types of location-based applications used by volunteer participants to better understand human, social, or physical behaviors.

Since the release of the Android operating system, Google has investigated a lot of efforts in the development of location-aware applications. Among these application, we can notice *Google Latitude* [20] and *Google MyTrack* [21]. *Google Latitude* is a location-aware mobile application developed by Google, which allows mobile users to track their location. The user's cell phone location is mapped on Google Maps and can control the accuracy and details of what each of the other users can see—an exact location can be allowed, or it can be limited to identifying the city only. However, Latitude overwrites a user's previous location with the new location data, and does not keep traces of locations provided to the service. *Google MyTrack* is another application for Android smartphones recording GPS tracks, while hiking, biking, running or participating in other outdoor activities. Once recorded, users can share tracks, upload them to *Google Spreadsheets* and visualize them on *Google Maps*. Although *Google MyTrack* shares with ANTDROID the objective of collecting activity traces of mobile users, *Google My Track* focuses on individuals and does not provide any support for collecting the activity traces of a population as ANTDROID does.

**Sensor-based Tracking.** In the domain of wireless sensor networks, several experiences have already been deployed to study and better understand wildlife and human behaviors [28, 36]. For example, the TRASHTRACK project [43] uses tags, which are

attached to different types of trash so that these items can be followed through the city's waste management system, revealing the final journey of our everyday objects in a series of real time visualizations. The trash tag periodically measures its location and reports the data to the server via the cellular network by making use of GPS and CDMA cell-tower trilateration based on the QUALCOMM INGEO platform. Interestingly, the deployment of this experiment has involved more than 50 researchers and developers while we believe that a large part of the platform they developed could have been reused from another tracking experiment.

**Real-time Tracking.** Finally, PACHUBE [58] and SMARTBIKE [42] are examples of emerging domain of interests related to energy saving and sustainable development issues. The SMARTBIKE wheel's sensing unit is capturing the effort level and information about user's surroundings, including road conditions, carbon monoxide, NOx, noise, ambient temperature and relative humidity. The collected data can be shared with friends, or with the city thereby contributing to a fine-grained database of environmental information from which we can all benefit. Similarly, PACHUBE is a community-contributed scalable infrastructure for storing, sharing, and discovering realtime sensor, energy and environment data from objects, devices, and buildings around the world. Although ANTDROID did not initially focused on this topic, we strongly believe that it could be extended to support real-time data processing by making use of *Complex Event Processing* (CEP) and *Really Simple Syndication* (RSS) feeds.

Similarly, STARTRACK NEXT GENERATION [26] provides a software infrastructure for developing track-based applications, such as *Ride-Sharing Service* or *Personalized Driving Directions*. These applications cover scenarios where a single user data can be used to personalize her experience based on her habitual tracks, for applications such as personalized advertising, recommendation systems, and health monitoring. As already mentioned, ANTDROID does not focus on real-time exploitation of activity traces (or tracks) for the time being. Nonetheless, and thanks to the use of the SCA component model, the infrastructure we developed can be extended to integrate track mining and comparison heuristics, which are used to adapt dynamically the behavior of user applications when similar user behaviors are detected. One potential application of this technology would target mobile social networks and would help people regularly sharing transportations to meet and discuss.

# 7 Conclusion

While it has been generally acknowledged as a keystone for the mobile computing community, the development of mobile sensing platforms remains a sensitive and critical task, which requires to take into account a variety of requirements covering both technical and ethical issues.

To address these challenges, we report in this paper on the design and the implementation of the ANTDROID distributed platform. This platform distinguishes between two roles: *scientists* requiring a sustainable environment to deploy sensing experiments and *participants* using their mobile device to contribute to scientific experiments. On the server-side, ANTDROID is built on the principles of Cloud computing and offers a modular service-oriented architecture, which can be customized upon scientist requirements. On the client-side, the ANTDROID platform provides a mobile application allowing to download experiments, running them in a dedicated sandbox and uploading datasets collected to the ANTDROID server.

Based on the principle of *only collect what you need*, the ANTDROID platform delivers an efficient yet flexible solution to ease the retrieval of realistic datasets. Beyond that, the following table summarizes the ethical and technical challenges we identified in this paper and reports on the contributions of the ANTDROID platform with regards to these key issues.

| **Challenges** | **ANTDROID Client** | **ANTDROID Server** |
|---|---|---|
| Privacy management | *privacy manager* | *privacy filters* |
| Sensing control | *privacy manager* | |
| User acceptation | *rewarding credits* | *web environment* |
| Energy consumption | *battery manager* | |
| Experiment deployment | | *experiment publisher* |
| Platform customization | | *SCA components* |

We therefore believe that ANTDROID can provide a sustainable foundation for building a new generation of context-aware services by leveraging the collection of realistic activity traces of mobile users. In particular, we reported on the exploitation of ANTDROID for *i)* testing location-based reactive systems and *ii)* building accurate mobility models.

In the future, we plan not only to extend the use cases covered by ANTDROID (*e.g.*, by studying epidemic propagation models like the flue), but also to improve the support for privacy and security by protecting the platform against malicious attacks. We are also investigating the multi-tenant nature of this platform and its impact on the flexibility and the scalability of the server infrastructure. Finally, we are be interested in supporting real-time processing of collected datasets in order to provide valuable feedbacks to the participants involved in an experiment (*e.g.*, by encouraging participants to explore uncovered areas in their neighborhood).

# References

[1] Beisiegel, M. et al. Service Component Architecture. http://www.osoa.org, 2007.

[2] Sonia Ben Mokhtar and Licia Capra. From Pervasive to Social Computing: Algorithms and Deployments. In *Int. Conf. on Pervasive Services*, pages 169–178. ACM, 2009.

[3] C. Bettini, S. Mascetti, X.S. Wang, and S. Jajodia. Anonymity in Location-based Services: Towards a General Framework. In *Int. Conf. on Mobile Data Management*, pages 69–76. IEEE, 2007.

[4] James Biagioni, Tomas Gerlich, Timothy Merrifield, and Jakob Eriksson. EasyTracker: automatic transit tracking, mapping, and arrival time prediction using smartphones. In *9th Int. Conf. on EmbeddedNetworked Sensor Systems*, pages 68–81. ACM, November 2011.

[5] S. Boag, D. Chamberlin, M.F. Fernández, D. Florescu, J. Robie, J. Siméon, and M. Stefanescu. Xquery 1.0: An xml query language. *W3C working draft*, 12, 2003.

[6] J.A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M.B. Srivastava. Participatory Sensing. 2006.

[7] H. Choi, S. Chakraborty, M. Greenblatt, Z.M. Charbiwala, and M.B. Srivastava. Sensorsafe: Managing health-related sensory information with fine-grained privacy controls. Technical report, Technical Report, September 2010.(TR-UCLA-NESL-201009-01), 2010.

[8] S. Consolvo, D.W. McDonald, T. Toscos, M.Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, et al. Activity Sensing in the Wild: a Field Trial of Ubifit Garden. In *26th Annual SIGCHI Conf. on Human factors in Computing Systems*, pages 1797–1806. ACM, 2008.

[9] D. Cuff, M. Hansen, and J. Kang. Urban Sensing: Out of the Woods. *Communications of the ACM*, 51(3):24–33, 2008.

[10] T. Das, P. Mohan, V.N. Padmanabhan, R. Ramjee, and A. Sharma. Prism: Platform for Remote Sensing Using Smartphones. In *8th Int. Conf. on Mobile Systems, Applications, and Services*, pages 63–76. ACM, 2010.

[11] P. Dutta, P.M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff. Common Sense: Participatory Urban Sensing Using a Network of Handheld Air Quality Monitors. In *7th ACM Conf. on Embedded Networked Sensor Systems*, pages 349–350. ACM, 2009.

[12] William Enck, Peter Gilbert, Byung gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N. Sheth. TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones. In *USENIX Symp. on Operating Systems Design and Implementation*, 2010.

[13] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation forwarding. In *9th ACM Int. Symp. on Mobile ad hoc networking and computing*, pages 251–260. ACM, 2008.

[14] O. Etzion and P. Niblett. *Event Processing in Action*. Manning Publications Co., 2010.

[15] H. Falaki, R. Mahajan, and D. Estrin. SystemSens: a tool for monitoring usage in smartphone research deployments. In *6th ACM Int. Work on Mobility in the Evolving Internet Architecture*, 2011.

[16] J. Froehlich, M.Y. Chen, S. Consolvo, B. Harrison, and J.A. Landay. Myexperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *5th Int. Conf. on Mobile Systems, Applications, and Services*, pages 57–70. ACM, 2007.

[17] S. Gambs, M.O. Killijian, and M.N. del Prado Cortez. Show Me How You Move and I Will Tell You Who You Are. In *3rd ACM SIGSPATIAL Int. Work. on Security and Privacy in GIS and LBS*, pages 34–41. ACM, 2010.

[18] S. Gambs, M.O. Killijian, and M. Nunez del Prado. GEPETO: a GEo-Privacy Enhancing Toolkit. In *Int. Work. on Advances in Mobile Computing and Applications: Security, Privacy and Trust*, 2010.

[19] S. Gambs, M.O. Killijian, and M. Nunez del Prado. Show me how you move and I will tell you who you are. In *3rd ACM SIGSPATIAL Int. Work. on Security and Privacy in GIS and LBS*, 2010.

[20] Google. Google Latitude, 2010. `http://www.google.com/latitude`.

[21] Google. Google MyTrack, 2010. `http://mytracks.appspot.com`.

[22] GoogleCode. Android Scripting (SL4A), 2010. `http://code.google.com/p/android-scripting`.

[23] C. Grün, S. Gath, A. Holupirek, and M. Scholl. XQuery Full Text Implementation in BaseX. *Database and XML Technologies*, pages 114–128, 2009.

[24] Christian Grün, Sebastian Gath, Alexander Holupirek, and Marc H. Scholl. XQuery Full Text Implementation in BaseX. In *6th Int. XML Database Symp. on Database and XML Technologies*, volume 5679 of *LNCS*, pages 114–128. Springer, 2009.

[25] M. Gruteser and D. Grunwald. Anonymous Usage of Location-based Services Through Spatial and Temporal Cloaking. In *1st Int. Conf. on Mobile Systems, Applications, and Services*, pages 31–42. ACM, 2003.

[26] Maya Haridasan, Iqbal Mohomed, Doug Terry, Chandramohan A. Thekkath, and Li Zhang. StarTrack Next Generation: A Scalable Infrastructure for Track-Based Applications. In *USENIX Symp. on Operating Systems Design and Implementation*, 2010.

[27] A. Holupirek, C. Grün, and M.H. Scholl. Basex & deepfs joint storage for filesystem and database. In *12th Int. Conf. on Extending Database Technology: Advances in Database Technology*, pages 1108–1111. ACM, 2009.

[28] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. *SIGARCH Comput. Archit. News*, 30(5):96–107, 2002.

[29] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic Monitoring and Accident Detection at Intersections. *Intelligent Transportation Systems, IEEE Transactions on*, 1(2):108–118, 2000.

[30] Marc-Olivier Killijian, Matthieu Roy, and Gilles Trédan. Beyond San Fancisco Cabs : Building a *-lity Mining Dataset. In *Work. on the Analysis of Mobile Phone Networks*, 2010.

[31] Minkyong Kim, David Kotz, and Songkuk Kim. Extracting a mobility model from real user traces. In *25th Annual Joint Conf. of the IEEE Computer and Communications Societies*, 2006.

[32] David Kotz and Tristan Henderson. Community Resource for Archiving Wireless Data At Dartmouth (CRAWDAD). http://crawdad.cs.dartmouth.edu.

[33] J.R. Kwapisz, G.M. Weiss, and S.A. Moore. Activity Recognition Using Cell Phone Accelerometers. *ACM SIGKDD Explorations Newsletter*, 12(2):74–82, 2011.

[34] N.D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A.T. Campbell. A Survey of Mobile Phone Sensing. *IEEE Communications Magazine*, 48(9):140–150, 2010.

[35] Lin Liao, Dieter Fox, and Henry A. Kautz. Location-Based Activity Recognition using Relational Markov Networks. In *IJCAI*, pages 773–778, 2005.

[36] Anders Lindgren, Cecilia Mascolo, Mike Lonegan, and Bernie Mcconnell. Seal2Seal: A Delay-Tolerant Protocol for Contact Logging in Wildlife Monitoring Sensor Networks. In *IEEE Int. Conf. on Mobile Ad-hoc and Sensor Systems*, 2008.

[37] L. Liu, C. Andris, A. Biderman, and C. Ratti. Uncovering Taxi Driver's Mobility Intelligence through His Trace. *IEEE Pervasive Computing*, 2009.

[38] D.C. Luckham. *Power of Events: an Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., 2001.

[39] E. Miluzzo, N.D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S.B. Eisenman, X. Zheng, and A.T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *6th ACM Conf. on Embedded Network Sensor Systems*, pages 337–350. ACM, 2008.

[40] E. Miluzzo, N.D. Lane, H. Lu, and A.T. Campbell. Research in the App Store Era: Experiences from the CenceMe App Deployment on the iPhone. In *1st Int. Work. Research in the Large: Using App Stores, Markets, and other wide distribution channels in UbiComp research*, 2010.

[41] MIT Media Lab. Reality Mining. http://reality.media.mit.edu.

[42] MIT SENSEable City lab. Copenhagen Wheel. http://senseable.mit.edu/copenhagenwheel, 2009.

[43] MIT SENSEable City lab. TRASHTRACK. http://senseable.mit.edu/trashtrack, 2009.

[44] P. Mohan, V.N. Padmanabhan, and R. Ramjee. Nericell: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones. In *6th ACM Conf. on Embedded network sensor systems*, pages 323–336. ACM, 2008.

[45] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda. PEIR, The Personal Environmental Impact Report, as a Platform for Participatory Sensing Systems Research. In *7th Int. Conf. on Mobile Systems, Applications, and Services*, pages 55–68. ACM, 2009.

[46] L. Nachman, A. Baxi, S. Bhattacharya, V. Darera, P. Deshpande, N. Kodalapura, V. Mageshkumar, S. Rath, J. Shahabdeen, and R. Acharya. Jog falls: A pervasive Healthcare Platform for Diabetes Management. *Pervasive Computing*, pages 94–111, 2010.

[47] Orange Labs R&D and faberNovel. Urban Mobs. http://www.urbanmobs.fr.

[48] Mika Raento, Antti Oulasvirta, Renaud Petit, and Hannu Toivonen. ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications. *IEEE Pervasive Computing*, 4(2):51–59, 2005.

[49] Matthieu Roy and Marc-Olivier Killijian. Brief Announcement: A Platform for Experimenting with Mobile Algorithms in a Laboratory. In *28th Annual ACM Symp. on Principles of Distributed Computing*, pages 316–317. ACM, 2009.

[50] San Francisco. DataSF App Showcase. http://datasf.org/showcase.

[51] Lionel Seinturier, Philippe Merle, Damien Fournier, Nicolas Dolet, Valerio Schiavoni, and Jean-Bernard Stefani. Reconfigurable SCA Applications with the FRASCATI Platform. In *IEEE Int. Conf. on Services Computing*, 2009.

[52] J. Sharkey. Coding for life–battery life, that is. In *Google IO Developer Conf.*, 2009.

[53] C. Shepard, A. Rahmati, C. Tossell, L. Zhong, and P. Kortum. LiveLab: measuring wireless networks and smartphone users in the field. *ACM SIGMETRICS Performance Evaluation Review*, 38(3):15–20, 2011.

[54] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos. AnonySense: A System for Anonymous Opportunistic Sensing. *Pervasive and Mobile Computing*, 2010.

[55] Timothy Sohn, Alex Varshavsky, Anthony LaMarca, Mike Y. Chen, Tanzeem Choudhury, Ian E. Smith, Sunny Consolvo, Jeffrey Hightower, William G. Griswold, and Eyal de Lara. Mobility Detection Using Everyday GSM Traces. In *8th Int. Conf. on Ubiquitous Computing*, volume 4206 of *LNCS*, pages 212–224. Springer, 2006.

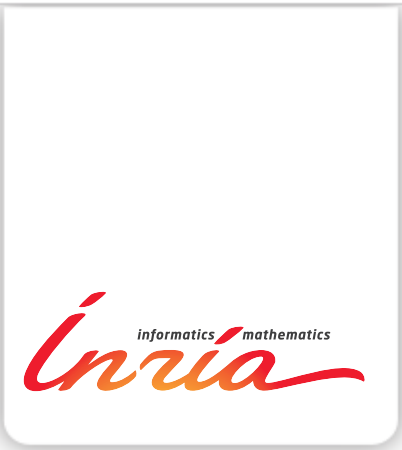
[56] Stanford University. Stanford University Mobile Activity TRAces (SUMATRA). http://infolab.stanford.edu/pleiades/SUMATRA.html.

[57] United States Government. Data.gov. http://www.data.gov.

[58] Usman Haque. PACHUBE. http://www.pachube.com.

[59] A. Vahdat, D. Becker, et al. Epidemic routing for partially connected ad hoc networks. Technical report, Technical Report CS-200006, Duke University, 2000.

# Contents




**RESEARCH CENTRE**
**LILLE – NORD EUROPE**

Parc scientifique de la Haute-Borne
40 avenue Halley - Bât A - Park Plaza
59650 Villeneuve d'Ascq

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399